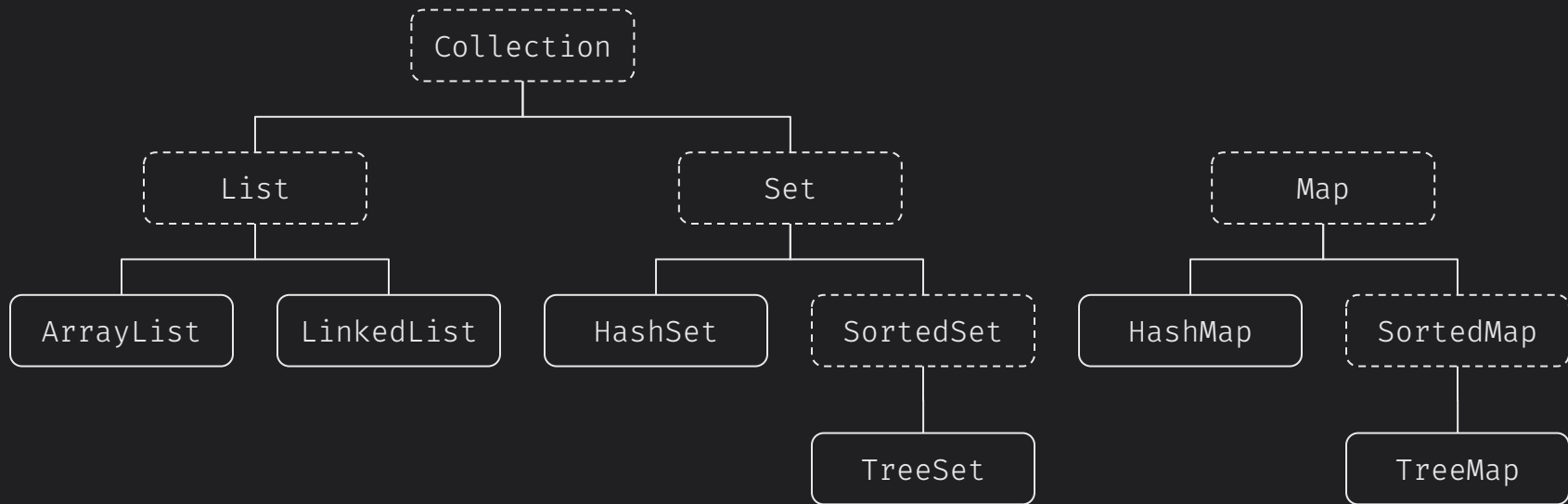


Collections Revisited

CS 272 Software Development

Collections Framework*



*Simplified Framework

Module java.base

Package java.util

Class ArrayList<E>

```
java.lang.Object
  java.util.AbstractCollection<E>
    java.util.AbstractList<E>
      java.util.ArrayList<E>
```

Type Parameters:

E - the type of elements in this list

All Implemented Interfaces:

Serializable, Cloneable, Iterable<E>, Collection<E>, List<E>, RandomAccess

Direct Known Subclasses:

AttributeList, RoleList, RoleUnresolvedList

```
public class ArrayList<E>
  extends AbstractList<E>
  implements List<E>, RandomAccess, Cloneable, Serializable
```

<https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/util/ArrayList.html>



Module java.base

Package java.util

Class ArrayList<E>

```
java.lang.Object
  java.util.AbstractCollection<E>
    java.util.AbstractList<E>
      java.util.ArrayList<E>
```

Type Parameters:

E - the type of elements in this list

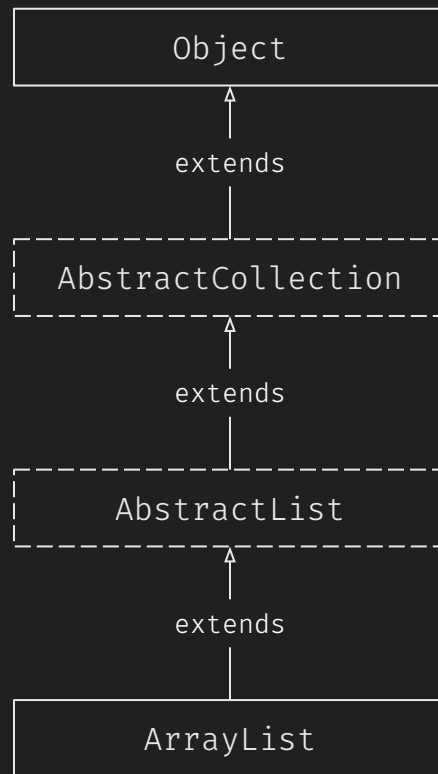
All Implemented Interfaces:

Serializable, Cloneable, Iterable<E>, Collection<E>, List<E>, RandomAccess

Direct Known Subclasses:

AttributeList, RoleList, RoleUnresolvedList

```
public class ArrayList<E>
  extends AbstractList<E>
  implements List<E>, RandomAccess, Cloneable, Serializable
```



Module java.base

Package java.util

Class ArrayList<E>

java.lang.Object

java.util.AbstractCollection<E>

java.util.AbstractList<E>

java.util.ArrayList<E>

Type Parameters:

E - the type of elements in this list

All Implemented Interfaces:

Serializable, Cloneable, Iterable<E>, Collection<E>, List<E>, RandomAccess

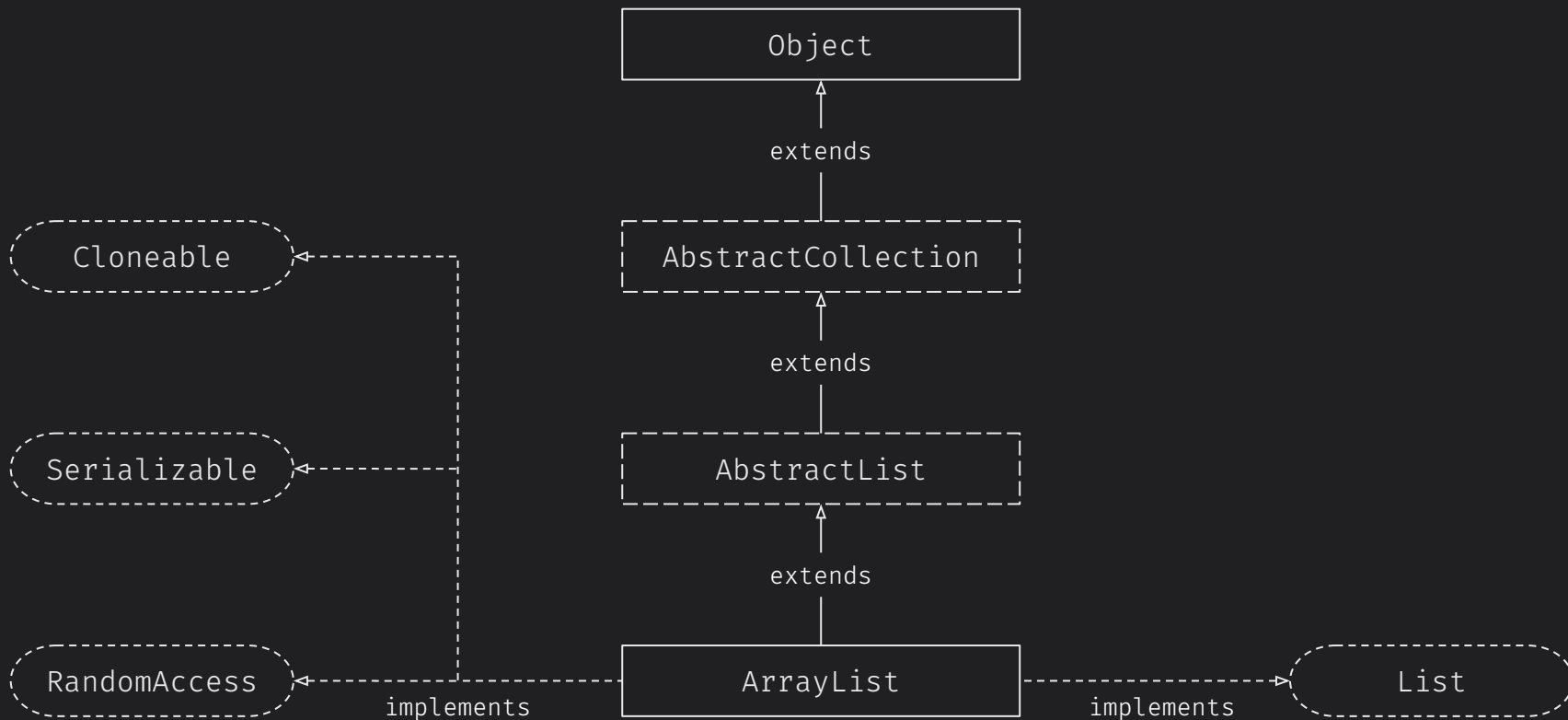
Direct Known Subclasses:

AttributeList, RoleList, RoleUnresolvedList

```
public class ArrayList<E>  
    extends AbstractList<E>  
    implements List<E>, RandomAccess, Cloneable, Serializable
```

<https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/util/ArrayList.html>





Module java.base

Package java.util

Class ArrayList<E>

java.lang.Object

java.util.AbstractCollection<E>

java.util.AbstractList<E>

java.util.ArrayList<E>

Type Parameters:

E - the type of elements in this list

All Implemented Interfaces:

Serializable, Cloneable, Iterable<E>, Collection<E>, List<E>, RandomAccess

Direct Known Subclasses:

AttributeList, RoleList, RoleUnresolvedList

```
public class ArrayList<E>
```

```
extends AbstractList<E>
```

```
implements List<E>, RandomAccess, Cloneable, Serializable
```

<https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/util/ArrayList.html>



Module java.base

Package java.util

Interface List<E>

Type Parameters:

E - the type of elements in this list

All Superinterfaces:

Collection<E>, Iterable<E>

All Known Implementing Classes:

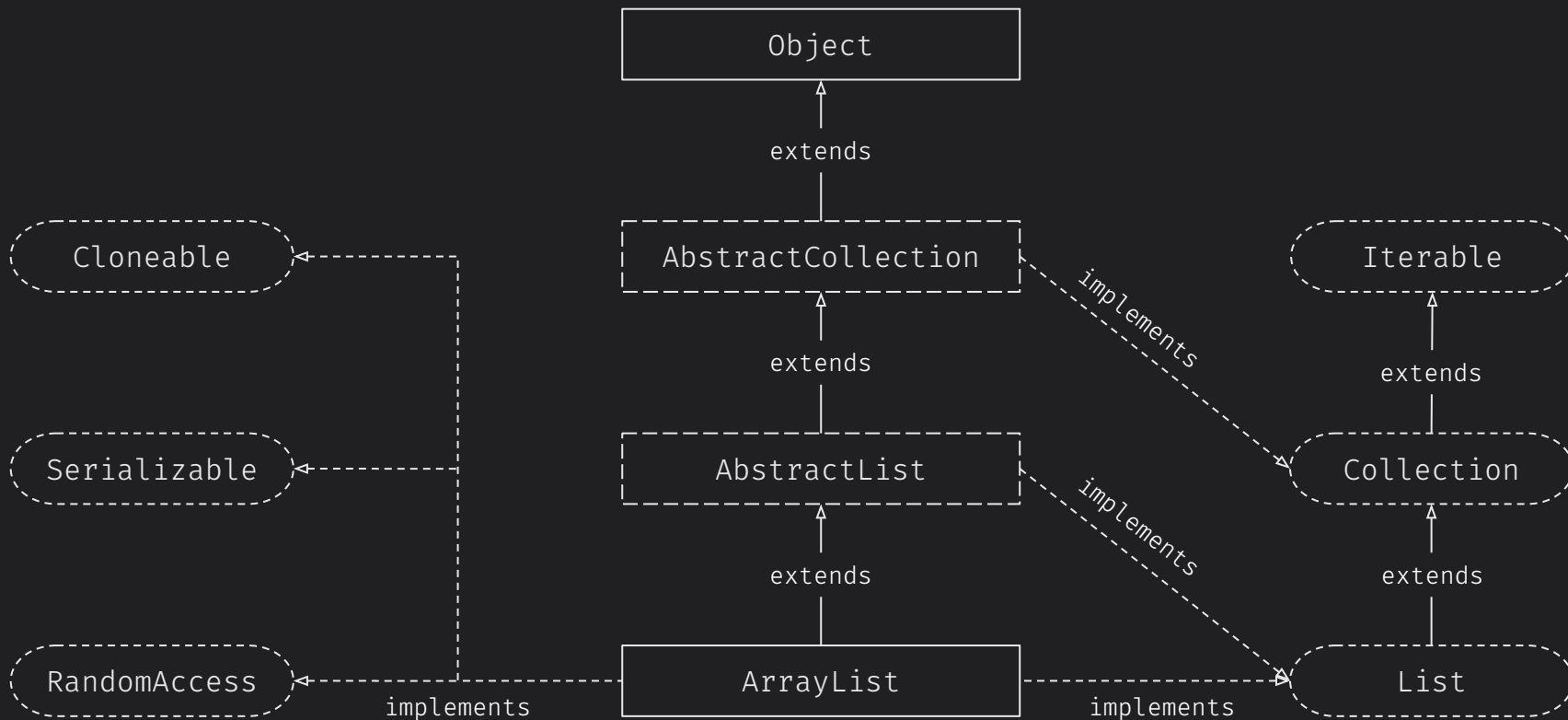
AbstractList, AbstractSequentialList, ArrayList, AttributeList, CopyOnWriteArrayList, LinkedList, RoleList, RoleUnresolvedList, Stack, Vector

```
public interface List<E>  
    extends Collection<E>
```

An ordered collection (also known as a *sequence*). The user of this interface has precise control over where in the list each element is inserted. The user can access elements by their integer index (position in the list), and search for elements in the list.

<https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/util/List.html>





ArrayList Revisited

- Interface **Collection** extends the **Iterable** interface
- Interface **List** extends **Collection**
- Class **AbstractCollection** implements **Collection**
- Class **AbstractList** extends **AbstractCollection** and implements **List** (and hence **Collection**)
- Class **ArrayList** extends **AbstractList** and implements others on top of **List** and **Collection**

<https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/util/ArrayList.html>



Collection Interface

- Root of collection hierarchy is an interface!
- Includes methods such as `add()`, `clear()`, `contains()`, `remove()`, `size()`, `toArray()`
- Method `iterator()` inherited from `Iterable`
 - Allows any collection to be used in for-each loops

<https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/util/Collection.html>



List Interface

- Extends `Collection` interface
 - And thus also inherits from `Iterable`
- Adds positional methods to `get`, `insert`, `modify`, or `remove` elements by position
- Adds ability to create a sublist

<https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/util/List.html>



AbstractCollection Class

- An abstract class that implements `Collection`
- Optional methods all throw an unsupported operation exception (discussed later)
- Provides skeleton implementations of other methods except `iterator()` and `size()`

<https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/util/AbstractCollection.html>



AbstractList Class

- An abstract class that extends `AbstractCollection` and implements `List` (and hence `Collection`)
- Optional methods still throw exceptions
- Provides iterator implementations for any list
- Provides skeletal implementations for all except `get()` and `size()` from `AbstractCollection`

<https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/util/AbstractList.html>



Unsupported Operations

- `Collections` class has methods to create **unmodifiable** versions of each collection type
- Throws `UnsupportedOperationException` to prevent modification operation
- Same exception thrown by implementations that do not support **optional** methods in hierarchy

<https://docs.oracle.com/en/java/javase/15/docs/api/java.base/java/lang/UnsupportedOperationException.html>

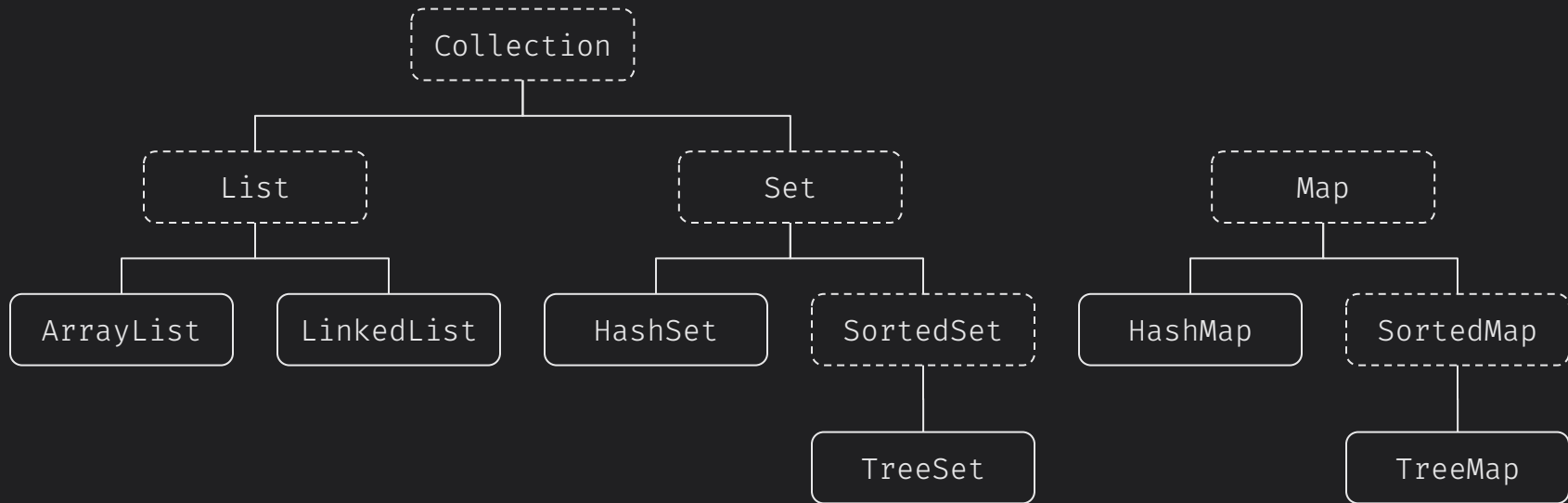


Abstract Classes

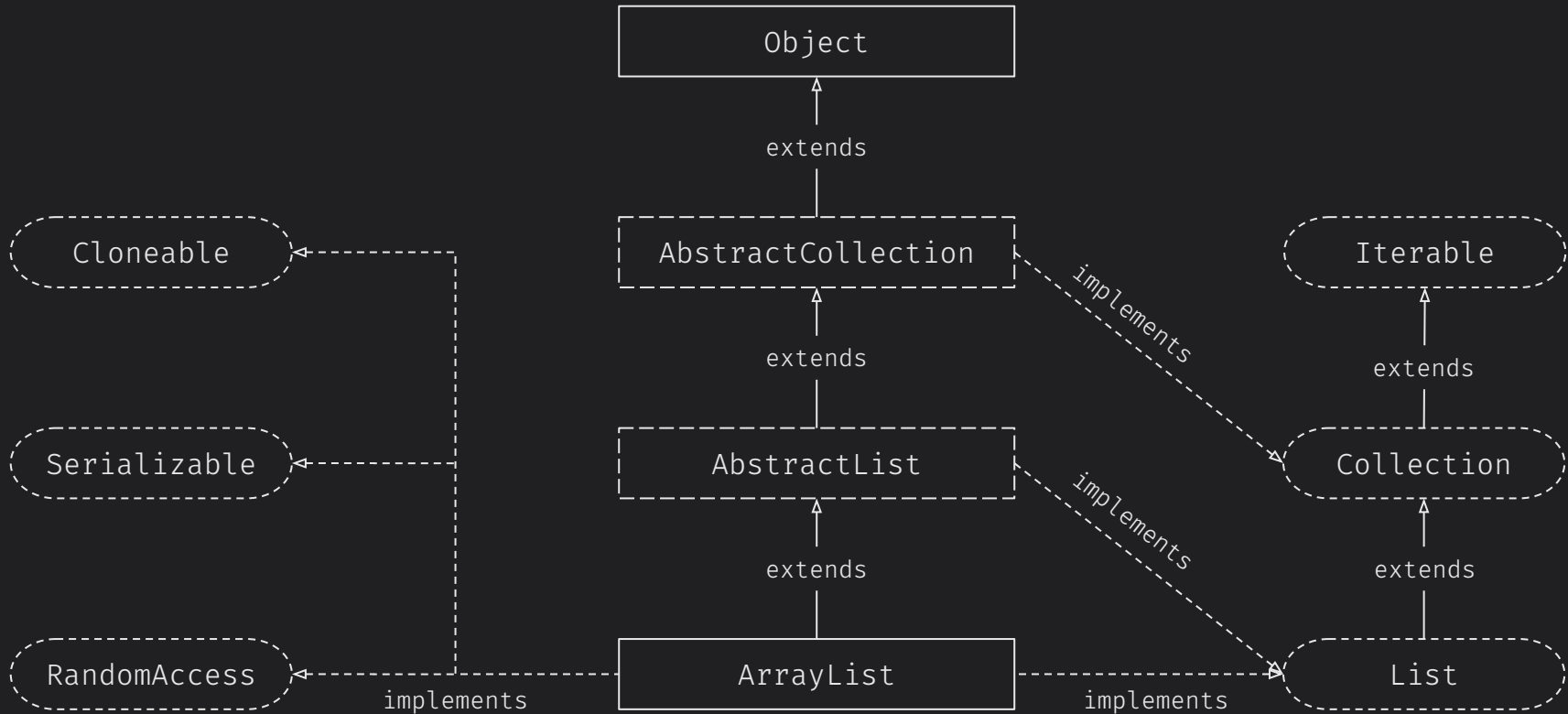
- Implement interfaces in Collection hierarchy and provide basic implementations where possible
- Includes AbstractCollection, AbstractMap, AbstractList, AbstractSequentialList, AbstractSet, and AbstractQueue
- Usually what is extended by actual implementations



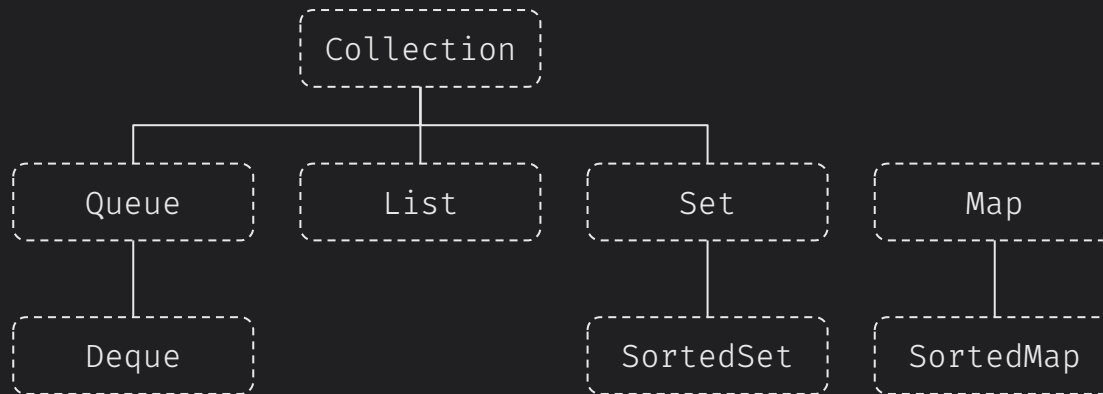
Collections Framework*



*Simplified Framework



Core Interface Hierarchy



<https://docs.oracle.com/javase/tutorial/collections/interfaces/index.html>

<https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/util/doc-files/coll-index.html>



Questions?

